Towards Analyzing Korean Learner Particles

Chong Min Lee, Soojeong Eom, and Markus Dickinson

AALL-09; March 10, 2009; Tempe, AZ
ICALL goal: provide intelligent feedback to learners on language production (cf. Heift and Schulze, 2007)
ICALL goal: provide intelligent feedback to learners on language production (cf. Heift and Schulze, 2007)

- 1st step: automatically assign linguistic analysis to sentence
ICALL goal: provide intelligent feedback to learners on language production (cf. Heift and Schulze, 2007)

- 1st step: automatically assign linguistic analysis to sentence
- Requires grammatical description of (in)appropriately-used constructions
  - e.g., subject-verb agreement
ICALL goal: provide intelligent feedback to learners on language production (cf. Heift and Schulze, 2007)

- 1st step: automatically assign linguistic analysis to sentence
- Requires grammatical description of (in)appropriately-used constructions
  - e.g., subject-verb agreement

Need to carefully consider the appropriate representation for a language to support the analysis of learner constructions
Idea: Use corpus annotation to build technology appropriate for distinctions learners know
Supporting feedback

Idea: Use corpus annotation to build technology appropriate for distinctions learners know

- Potentially saves time & effort
- Connects to state-of-the-art parsing (e.g., Charniak and Johnson, 2005; Nivre et al., 2007)
Supporting feedback

Idea: Use corpus annotation to build technology appropriate for distinctions learners know

- Potentially saves time & effort
- Connects to state-of-the-art parsing (e.g., Charniak and Johnson, 2005; Nivre et al., 2007)

But is corpus annotation appropriate for analyzing learner data?
Supporting feedback

Idea: Use corpus annotation to build technology appropriate for distinctions learners know

- Potentially saves time & effort
- Connects to state-of-the-art parsing (e.g., Charniak and Johnson, 2005; Nivre et al., 2007)

But is corpus annotation appropriate for analyzing learner data?

**Overarching Goal:** provide framework for re-using corpus annotation in a way which supports providing feedback
Dickinson and Lee (to appear) outline a framework for converting corpus annotation into an analysis that is desirable

- Promising initial results, but only initial results . . .
Modeling learner language

Dickinson and Lee (to appear) outline a framework for converting corpus annotation into an analysis that is desirable

- Promising initial results, but only initial results . . .

Goals for this work-in-progress:

1. Use a real learner corpus for evaluation
2. Adapt other NLP technology—namely, a POS tagger
3. Continue to develop parsing technology
Background: Korean particles

Korean postpositional particles indicate grammatical functions, thematic roles, and locations of people & objects

- Similar to English prepositions, but wider range of functions:

  (1) Sumi-\textit{neun} chaek-i pilyohae-yo
      Sumi-TOP book-SBJ need-polite
      ‘Sumi needs a book.’
Background: Korean particles

Korean postpositional particles indicate grammatical functions, thematic roles, and locations of people & objects

- Similar to English prepositions, but wider range of functions:

  (2) Sumi-\textit{neun} chaek-\textit{i} pilyohae-yo
      Sumi-TOP book-SBJ need-polite

      ‘Sumi needs a book.’

- Focus of ICALL systems for Korean & Japanese (Dickinson et al., 2008; Nagata, 1995)
Learners of Korean often misuse particles (Ko et al., 2004)

(3) *Sumi-neun chaek-*eul pilyohae-yo
   Sumi-TOP book-OBJ need-polite
   ‘Sumi needs a book.’
Learners of Korean often misuse particles (Ko et al., 2004)

(4) *Sumi-neun chaek-**eul** pilyohae-yo
   Sumi-TOP book-OBJ need-polite
   ‘Sumi needs a book.’

Lee et al. (to appear) & Ko et al. (2004) categorize particle errors by learners of Korean into 6 types; we focus on 2:

- *Omission & replacement* errors: 60%+ of particle errors made by beginning learners (Lee et al., to appear)
Usage of Korean particles

We focus on *syntactic* postpositional particles

- Case markers: indicate relationship between verb & noun

(5) Sumi-*ka* Jisu-*ege* chaek-*eul* ju-ass-*ta*.
    Sumi-SBJ Jisu-DAT book-OBJ give-PAST-DECL
    ‘Sumi gave Jisu a book.’
Usage of Korean particles

We focus on *syntactic* postpositional particles

- **Case markers**: indicate relationship between verb & noun

  (6) Sumi-*ka*  Jisu-*ege*  chaek-*eul*  ju-ass-*ta*.
  Sumi-SBJ  Jisu-DAT  book-OBJ  give-PAST-DECL

  ‘Sumi gave Jisu a book.’

- **Modifiers** (cf. prepositions): indicate specific lexical, syntactic, & semantic information between verb & noun
Parsing for learner language
What we have: constituencies

The data we use:
- Penn Korean Treebank (KTB), v. 2.0 (Han et al., 2002)
- Syntactically-annotated corpus with constituency annotation & function labels (e.g., subject (SBJ))
Parsing for learner language

What we want: dependencies

We want dependency structures

(7) toduk+i munseo+reul humchi+eo ka+ass+ta
    a burglar    a document    steal    go

Appropriate for Korean & Japanese (e.g., Chung, 2004; Seo, 1993; Kudo and Matsumoto, 2000).
Parsing for learner language

What we want: dependencies

We want dependency structures

(8) toduk+i munseo+reul humchi+eo ka+ass+ta
   a burglar   a document   steal     go

- Appropriate for Korean & Japanese (e.g., Chung, 2004; Seo, 1993; Kudo and Matsumoto, 2000).
- Dependency relations provide relevant feedback information
Limitations of current annotation

Dependency relations

Constituency-to-dependency conversion is straightforward (cf., e.g., Collins, 1999; Nilsson and Hall, 2005)

- But what dependency labels do we use?
Limitations of current annotation

Dependency relations

Constituency-to-dependency conversion is straightforward (cf., e.g., Collins, 1999; Nilsson and Hall, 2005)

- But what dependency labels do we use?

KTB has somewhat coarse function labels

- e.g., COMP realizable by several kinds of particles
Limitations of current annotation

Particle annotation

KTB has syntactic role particles PCA (case), PAD (adverbial), & PAN (adnominal)

- Each label realizable by several particles

(9) a. (NP-ADV naenyeon-e/PAD) boneos-reul batneunta
   next year+at bonus-OBJ receive

b. (NP-ADV naenyeon-buteo/PAD) boneos-reul batneunta
   next year+from bonus-OBJ receive

c. * (NP-ADV naenyeon-eso/PAD) boneos-reul batneunta
   next year+from bonus-OBJ receive
Recovering information from annotation
Including particle names

**Solution:** Put particle information into labels
Recovering information from annotation
Including particle names

**Solution:** Put particle information into labels

1. **Normalization:** group particles that function in same manner
   - their selection relies on non-syntactic factors
Recovering information from annotation
Including particle names

**Solution:** Put particle information into labels

1. **Normalization:** group particles that function in same manner
   - their selection relies on non-syntactic factors

<table>
<thead>
<tr>
<th>POS</th>
<th>Class</th>
<th>Particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>SBJ</td>
<td>-kkeseo, -seo, -ka/-i, -eseo</td>
</tr>
<tr>
<td></td>
<td>OBJ</td>
<td>-eul/-reul</td>
</tr>
<tr>
<td>PAN</td>
<td>UI</td>
<td>-ui</td>
</tr>
<tr>
<td>PAD</td>
<td>EUROSSEO</td>
<td>-eusosseo</td>
</tr>
<tr>
<td></td>
<td>EUROPUTEO</td>
<td>-europuteo</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>-e</td>
</tr>
<tr>
<td></td>
<td>EGE</td>
<td>-ege</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Recovering information from annotation
Including particle names

**Solution:** Put particle information into labels

1. **Normalization:** group particles that function in same manner
   - their selection relies on non-syntactic factors

<table>
<thead>
<tr>
<th>POS</th>
<th>Class</th>
<th>Particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>SBJ</td>
<td>-kkeseo, -seo, -ka/-i, -eseo</td>
</tr>
<tr>
<td></td>
<td>OBJ</td>
<td>-eul/-reul</td>
</tr>
<tr>
<td>PAN</td>
<td>UI</td>
<td>-ui</td>
</tr>
<tr>
<td>PAD</td>
<td>EUROSSEO</td>
<td>-eurosseo</td>
</tr>
<tr>
<td></td>
<td>EUROPUTEO</td>
<td>-europuteo</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>-e</td>
</tr>
<tr>
<td></td>
<td>EGE</td>
<td>-ege</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

2. **Threshold:** focus on particles appearing $> 50$ times in corpus

Chong Min Lee, Soojeong Eom, and Markus Dickinson
Towards Analyzing Korean Learner Particles
Removing information from annotation

But isn’t this highly redundant?

- e.g., EGE will be used whenever ege is encountered
Removing information from annotation

But isn’t this highly redundant?

  ■ e.g., EGE will be used whenever ege is encountered

However: Labels with particle names predict the presence of particular (type of) particle, even if that particle is not there
Removing information from annotation

But isn’t this highly redundant?

- e.g., EGE will be used whenever ege is encountered

**However:** Labels with particle names predict the presence of particular (type of) particle, even if that particle is not there

- **Idea:** Remove redundancy for *a second model* by removing particles from word forms
Removing information from annotation

But isn’t this highly redundant?

- e.g., EGE will be used whenever ege is encountered

**However:** Labels with particle names predict the presence of particular (type of) particle, even if that particle is not there

- **Idea:** Remove redundancy for *a second model* by removing particles from word forms

- Parsing disagreements between models provide platform for error detection (cf. Metcalf and Boyd, 2006)
  - Shows success on artificially-created errors in news text
Adapting a learner corpus for evaluation

So far: Evaluated on artificial errors
Adapting a learner corpus for evaluation

So far: Evaluated on artificial errors

Next step: Use a Korean learner corpus for evaluation

- annotated for particle errors (Lee et al., to appear)
To evaluate positives & negatives of error detection before fully moving to unaltered learner data, we make some changes:

1. Correct misspelled/malformed particles (error type 4)
2. Correct spacing errors in particles (type 6) e.g., particles split from words are merged
3. Fix incorrect sentence boundaries
4. Tokenize punctuation separately
To evaluate positives & negatives of error detection before fully moving to unaltered learner data, we make some changes:

1. Correct misspelled/malformed particles (error type 4)
   - Other words are not corrected, to keep the data more real
Learner corpus changes (1)

Data compatibility

To evaluate positives & negatives of error detection before fully moving to unaltered learner data, we make some changes:

1. Correct misspelled/malformed particles (error type 4)
   - Other words are not corrected, to keep the data more real

2. Correct spacing errors in particles (type 6)
   - e.g., particles split from words are merged
Learner corpus changes (1)
Data compatibility

To evaluate positives & negatives of error detection before fully moving to unaltered learner data, we make some changes:

1. Correct misspelled/malformed particles (error type 4)
   - Other words are not corrected, to keep the data more real
2. Correct spacing errors in particles (type 6)
   - e.g., particles split from words are merged
3. Fix incorrect sentence boundaries
Learner corpus changes changes (1)

Data compatibility

To evaluate positives & negatives of error detection before fully moving to unaltered learner data, we make some changes:

1. Correct misspelled/malformed particles (error type 4)
   - Other words are not corrected, to keep the data more real
2. Correct spacing errors in particles (type 6)
   - e.g., particles split from words are merged
3. Fix incorrect sentence boundaries
4. Tokenize punctuation separately
We do not deal with discourse-based errors: honorifics & topics
We do not deal with discourse-based errors: honorifics & topics

- Discourse-based errors can occur within the error types we investigate (substitutions, omissions)
Learner corpus changes

Fine-grained annotation

We do not deal with discourse-based errors: honorifics & topics

- Discourse-based errors can occur within the error types we investigate (substitutions, omissions)

How can we properly evaluate our system on lexical case errors?
We do not deal with discourse-based errors: honorifics & topics

- Discourse-based errors can occur within the error types we investigate (substitutions, omissions)

How can we properly evaluate our system on lexical case errors?

**Solution:** Add error subtype information to the surface-level annotation scheme of Lee et al. (to appear)

- Indicate if error is honorific-based or topic-based
Adapt a POS tagger

**So far:** Used POS tags from the corpus
Adapt a POS tagger

**So far:** Used POS tags from the corpus

**Next step:** Use POS tagger for Korean (Han and Palmer, 2004)

- Based on same corpus tagset
- Good performance
  - Precision: 95.43%
  - Recall: 95.04%
Adapt a POS tagger

**So far:** Used POS tags from the corpus

**Next step:** Use POS tagger for Korean (Han and Palmer, 2004)

- Based on same corpus tagset
- Good performance
  - Precision: 95.43%
  - Recall: 95.04%

But tagger is designed for regular language

- How well will the tagger work on learner language?
  - cf. Shih et al. (2000); van Rooy and Schäfer (2002)
Initial tagging vs. hand-cleaned results

New genre

Moving from one genre to another leads to tagging problems:

*jungkuk*/VV +*eo*/ECS

*hae*/NNC +*yo*/PAU

⇔

*jungkukeo*/NNC

⇔

*ha*/VV +*yo*/EFN

China+language

⇔

Chinese

Formal and informal registers

Tagger trained on formal newstext: uses

da

Learner data is informal: uses

yo, e.g., for

haeyo

*ha*/VV +*yo*/EFN

sun+particle

⇔

*ha*/VV +*yo*/EFN

to do+verb-ending
Initial tagging vs. hand-cleaned results

New genre

Moving from one genre to another leads to tagging problems:

- Unknown words lead to mis-segmentation & mis-tagging

(12) *jungkuk/VV+eo/ECS ⇔ jungkukeo/NNC

China+language ≡ Chinese
Initial tagging vs. hand-cleaned results

New genre

Moving from one genre to another leads to tagging problems:

- Unknown words lead to mis-segmentation & mis-tagging

\[(14) \quad *\text{jungkuk/}VV+eo/ECS \leftrightarrow \text{jungkukeo/}NNC\]
\[\text{China} + \text{language} \quad \text{Chinese}\]

- Formal and informal registers
  - Tagger trained on formal newstext: uses *da* ending
  - Learner data is informal: uses *yo* ending, e.g., for *haeyo*:

\[(15) \quad *\text{hae/}NNC+yo/PAU \leftrightarrow \text{ha/}VV+yo/EFN\]
\[\text{sun+particle} \quad \text{to do+verb-ending}\]
Initial tagging vs. hand-cleaned results

Underlying forms

Tagger mishypothesizes underlying form (needed for feedback):
Initial tagging vs. hand-cleaned results

Underlying forms

Tagger mishypothesizes underlying form (needed for feedback):

- e.g., *deuleosseoyo* in a context to mean ‘listen’:

(17) *deul/VV+eoss/EPF+eoyo/EFN ⇔

  * lift+PAST+ENDING  
  * deud/VV+eoss/EPF+eoyo/EFN  
  * listen+PAST+ENDING
Steps for adapting the POS tagger

Current precision on hand-cleaned learner data:

- 72.0% (737/1024) (vs. 95% on regular language)
Steps for adapting the POS tagger

Current precision on hand-cleaned learner data:
- 72.0% (737/1024) (vs. 95% on regular language)

Based on this analysis of POS tagging errors, we intend to add a rule-based post-processing step which corrects for:
- Unknown word guessing errors
- Informal register
Preliminary error detection evaluation

To gauge current error detection, we:

1. POS tagged learner corpus
2.Parsed 2 versions of learner corpus (with/without particles)
3. Examined mismatches from parsing models

Results of using mismatches as heuristic to flag errors:

- Mismatches identify 765 out of 2655 positions
- Recall = 51.4% (54/105) (vs. 82.5% on artificial data)
- Recall indicates that mismatches can play a role as one piece of information for error detection
- Performance is similar without honorific/topic particles
Preliminary error detection evaluation

To gauge current error detection, we:

1. POS tagged learner corpus
2. Parsed 2 versions of learner corpus (with/without particles)
3. Examined mismatches from parsing models

Results of using mismatches as heuristic to flag errors:

- Mismatches identify 765 out of 2655 positions
- Recall = 51.4% (54/105) (vs. 82.5% on artificial data)
Preliminary error detection evaluation

To gauge current error detection, we:

1. POS tagged learner corpus
2. Parsed 2 versions of learner corpus (with/without particles)
3. Examined mismatches from parsing models

Results of using mismatches as heuristic to flag errors:

- Mismatches identify 765 out of 2655 positions
- Recall = 51.4% (54/105) (vs. 82.5% on artificial data)
  - Recall indicates that mismatches can play a role as one piece of information for error detection
Preliminary error detection evaluation

To gauge current error detection, we:

1. POS tagged learner corpus
2. Parsed 2 versions of learner corpus (with/without particles)
3. Examined mismatches from parsing models

Results of using mismatches as heuristic to flag errors:

- Mismatches identify 765 out of 2655 positions
- Recall = 51.4% (54/105) (vs. 82.5% on artificial data)
  - Recall indicates that mismatches can play a role as one piece of information for error detection

Performance is similar without honorific/topic particles
Error detection evaluation

Problems for current technology

We have not adapted our tools from news text to learner data
problems for current technology

we have not adapted our tools from news text to learner data

- there are multiple errors in a sentence, leading to low recall:
  - both models frequently provide no relevant label
Problems for current technology

We have not adapted our tools from news text to learner data

- There are multiple errors in a sentence, leading to low recall:
  - Both models frequently provide no relevant label
- Unknown words are a big problem
  - When neither verb nor noun is known, it is hard to guess the argument relations for a model without particles
We have not adapted our tools from news text to learner data

- There are multiple errors in a sentence, leading to low recall:
  - Both models frequently provide no relevant label
- Unknown words are a big problem
  - When neither verb nor noun is known, it is hard to guess the argument relations for a model without particles

**Next step:** Address problems by training on wider range of data

- We want to train the parser on Sejong corpus (Kim, 2005)
Summary and Outlook

Summary:

- Examined how to provide parsing model for information about Korean postpositional particles
  - Identified challenges & opportunities for using POS tagger
  - Began to evaluate on learner data
- Highlighted the need to add more syntactically-annotated data

Outlook:

- Extend the parser to handle a wider range of data
- Integrate tools into a more robust error detection module (cf., e.g., Tetreault and Chodorow, 2008)
- Use dependency labels to perform error diagnosis in a real ICALL setting (Dickinson et al., 2008)
Summary and Outlook

Summary:
- Examined how to provide parsing model for information about Korean postpositional particles
  - Identified challenges & opportunities for using POS tagger
  - Began to evaluate on learner data
- Highlighted the need to add more syntactically-annotated data

Outlook:
- Extend the parser to handle a wider range of data
- Integrate tools into a more robust error detection module (cf., e.g., Tetreault and Chodorow, 2008)
- Use dependency labels to perform error diagnosis in a real ICALL setting (Dickinson et al., 2008)
Acknowledgements

Our thanks to:

- Sun-Hee Lee & SeokBae Jang for providing their learner corpus
- Ross Israel for general work & insights
- Rebecca Sachs & Yunkyoung Kang for support on Korean ICALL
- Members of the IU autumn 2009 L700 seminar for feedback on this general line of research


Han, Chung-Hye, Na-Rare Han, Eon-Suk Ko and Martha Palmer (2002). Development and Evaluation of a Korean Treebank and its Application to NLP. In *Proceedings of LREC-02*.


Nivre, Joakim, Johan Hall, Jens Nilsson, Atanas Chanev, Gulsen Eryigit, Sandra Chong Min Lee, Soojeong Eom, and Markus Dickinson Towards Analyzing Korean Learner Particles


